UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/713,405	11/14/2003	Bryan M. Cantrill	03226.350001; SUN040252	7015

32615 7590 07/30/2007

OSHA LIANG L.L.P./SUN
1221 MCKINNEY, SUITE 2800
HOUSTON, TX 77010

EXAMINER	
VO, TED T	

ART UNIT	PAPER NUMBER
2191	

MAIL DATE	DELIVERY MODE
07/30/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/713,405	CANTRILL, BRYAN M.	
	Examiner	Art Unit	
	Ted T. Vo	2191	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *09 May 2007*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-38* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-38* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. This action is in response to the communication filed on 05/09/2007.

Claims 1-38 are pending in the application.

Response to Arguments

2. The arguments given in Remarks, filed on 05/09/07 have been considered but not persuasive.

It should be noted that a claimed invention in which its effort is for changing size, shape, adding ingredients, making portable, making integral, making separable, rearrangement of elements, etc., over a prior art does not cause any patentable differences over the prior art (In re Rose , 220 F.2d 459, 105 USPQ 237 (CCPA 1955); In re Dailey, 357 F.2d 669, 149 USPQ 47 (CCPA 1966); In re Japikse, 181 F.2d 1019, 86 USPQ 70 (CCPA 1950), etc...).

In the remarks, the arguments appear not to discuss what the patentable differences are, but to attack the language specific used within the reference and the claims. For example, the arguments alleged that

> "Specifically, the Examiner has asserted that "instrumentation code" is equivalent to program "source code." *See* Office Action mailed February 9, 2007, p. 2. By asserting that "instrumentation code" is equivalent to program "source code," the Examiner is completely misconstruing the broadest ordinary meaning of the term program "source code.""

Regarding the reference, Tamches provides a method for defining trace points in "a program" (i.e. a program per se) such as the programs seen Figure 3.3, p. 33, Figure 4.9, p. 62, etc. Yes, a skill of the art like Tamches directs the program to a specific kernel program, but like any invention application, its principle and its method do not limit to only the kernel program. A skill in the art knows that any program is applicable by the Tamches' method for defining trace points, particularly at a branch.

It should be noted that a kernel program or a generic program is only a program per se. Defining trace points on a kernel program or defining a trace point on a generic program falls into the cases addressed by the court.

In the reference, Figure 3.3, p. 33, depicts an example for how a trace point would be defined. Clearly, it does not make any difference to a generic program. The program A, B, or C might be a kernel program, but its program structure is a generic program. It has branches, has call instructions, and it is editable by users for entering trace points; it requires compiling, optimizing, before transformed into machine code for execution.

The arguments fail to address any patentable advantage over this reference. The claims attempt covering any aspect including the method done by Tamches. If there was any difference, it would be similar to the court' rules in MPEP, i.e. changing size, shape, adding ingredients, making portable, making integral, making separable, rearrangement of elements, etc.

Claim Rejections - 35 USC § 102

3. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

4. Claims 1-38 are rejected under 35 U.S.C. 102(b) as being anticipated by Tamches, "Fine-Grained Dynamic Instrumentation of Commodity Operating System Kernels", University of Wisconsin, Pages: 1-141, 2001.

Given the broadest reasonable interpretation of followed claims in light of the specification.

As per Claim 1: Tamches discloses,

A method of defining a trace point, comprising:

defining a trace point representation in a program source code (See p. 49, for example, Figure 4,1, instrumentation point, where *a program source code* is instrumentation code discussed in p. 15, first full paragraph)*;*

compiling the program source code to generate an instrumented program (See p. 15, first full paragraph, instrumentation code is compiled), ***comprising the trace point corresponding to the trace point representation; and***

associating the trace point with a placeholder function (splicing: See Chapter 4, where splicing is used to joint patch code at a defined instrumentation point) ***configured to produce a minimal disabled probe effect.*** (See p. 49, place code patch at a branch will *produce a minimal disabled probe effect*).

As per Claim 2: Tamches discloses, ***The method of claim 1, further comprising:***

storing an address location (returning jump, p. 51:1-9) ***of the trace point and a probe handler*** (Figure 3.1) ***associated with the address location in a trace point table*** (a set of instrumentation points, allocated)***, wherein the address location and the probe handler are identified by a trace point identifier*** (see p. 66, within sec 4.6.1)

See Chapter 4, also see discussing trap handler in p. 15, including Kitrace, and IBM Dprobes.

As per Claim 3: Tamches discloses, ***The method of claim 2, further comprising:***

obtaining the address location of the trace point; and

identifying the probe handler associated with the address location. (p. 55, see sec 4.3, Splicing: Jumping to the Code Patch)

As per Claim 4: Tamches discloses, ***The method of claim 1, wherein the trace point representation comprises a tracing function defined by a code*** (function defined by code patch).

As per Claim 5: Tamches discloses, ***The method of claim 1, wherein the placeholder function comprises at least one instruction designed to use minimal system resources*** (p. 51, see sec. 4.1, dealt with available scratch registers).

As per Claim 6: Tamches discloses, ***The method of claim 5, wherein at least one instruction comprises a no-operation instruction*** (refer to nop instruction seen in the reference; e.g., in p. 53).

As per Claim 7: Tamches discloses, ***The method of claim 5, wherein the at least one instruction comprises a first instruction comprising a trap instruction*** (discussing save a trap, sec. 4.1, p. 50) ***and a second instruction comprising a no-operation instruction*** (discussing a nop instruction, p. 53).

As per Claim 8: Tamches discloses, ***The method of claim 2, further comprising:***

obtaining a tracing function name from trace object code using a tracing framework (Figure 3.1, trace function such as code patch defined by a client);

determining an address location of the trace point in the instrumented program by accessing the trace function name in the trace point table (the instrumentation points, allocated by using splicing)***; and replacing a placeholder function located at the address location of trace point with a function call into the tracing framework*** (splicing).

As per Claim 9: Tamches discloses, ***A method for enabling a trace point, comprising:***

obtaining a tracing function name from trace object code using a tracing framework (See Figure 3.1. p. 27; i9.e. an instrumentation point that causes jumping to a code patch (discussed in p. 67)), ***wherein the tracing function name comprises a probe handler*** (e.g. p. 66, last three lines of the page);

determining an address location of the trace point in an instrumented program by accessing the probe handler in a trace point table (p. 66, for example, a hash table that maps instrumentation point address/code patch)***; and***

replacing a placeholder function (Splicing, such as Figure 4.1, p. 49, that causes jumping to code patch) ***located at the address location of the trace point with a function call into the tracing framework*** (For example, see in Figure 4.1, address location is address where instrumentation point is assigned).

As per Claim 10: Tamches discloses, ***The method of claim 9, further comprising:***

disabling the trace point by replacing the function call with the placeholder function (i.e. a point after instrumentation point will not be executed until a return from the code patch).

As per Claim 11: Tamches discloses, ***The method of claim 9, further comprising:***

storing the probe handler and the address location in a trace point table, wherein the trace function name and address location are identified by a trace point identifier (splicing mechanism used for replacing instrumentation at entry points in an instrumented program).

As per Claim 12: Tamches discloses, ***The method of claim 9, further comprising:***

accessing a probe provider using the tracing framework; and directing the probe provider to enable a probe associated with the trace point (The mechanism of Figure 3.1. Note: Tamches discussed IBM Dprobes that does the functionality of the claim).

As per Claim 13: Tamches discloses, ***The method of claim 9, further comprising:***

generating the trace point table during compilation of a program source code (See in p. 26, the introduction to KernInst, using compiler's debugging).

As per Claim 14: Tamches discloses, ***The method of claim 9, wherein the function call comprises calling to a jump instruction to the tracing framework*** (Jump/branch are used in the KernInst and in instrumented programs).

As per Claim 15: Tamches discloses, ***The method of claim 9, wherein the function call comprises:***

generating a trap transferring control to a trap handler associated with the trap;

calling the tracing framework from the trap handler (See p. 13, discussion of Dynamic Instrumentation in User Programs, particularly on code splicing overwriting an instruction with trap)***; and***

emulating a patched-over instruction (See Figure 4.1).

As per Claim 16: Tamches discloses, ***The method of claim 15, wherein the emulating comprises incrementing a saved instruction pointer by one prior to issuing a return instruction from the trap*** (It is a mechanism of programming where "return" remembers where is the next pointer after control has been transferred, See Figure 4.1).

As per Claim 17: Tamches discloses, ***The method of claim 9, wherein the placeholder function comprises at least one instruction designed to use minimal system resources*** (Referred to the number of available scratch registers (already mentioned)).

As per Claim 18: Tamches discloses, ***The method of claim 17, wherein the at least one instruction comprises a no- operation instruction*** (refer to nop instruction in the reference).

As per Claim 19: Tamches discloses, ***The method of claim 17, wherein the at least one instruction comprising a first instruction comprises a trap instruction and a second instruction comprising a no-operation instruction*** (refer to nop instruction in the reference).

As per Claim 20: Tamches discloses, *A computer system on a network for defining a trace point comprising: a processor; a memory; a storage device;*

(computer per se)

and software instructions stored in the memory for enabling the computer system to:

define a trace point representation in a program source code;

compile the program source code to generate an instrumented program comprising the trace point corresponding to the trace point representation; and

wherein the trace point is associated with a placeholder function configured to produce a minimal disabled probe effect.

See rationale discussed in the rejection of claim 1 above.

As per Claims 21-27: Tamches discloses claims 21-27. See the rationale discussed in the rejection of Claims 2, 8, 3-7 above, respectively.

As per Claim 28: Tamches discloses, *A computer system on a network for enabling a trace point comprising: a processor; a memory; a storage device;*

(computer per se)

and software instructions stored in the memory for enabling the computer system to:

obtain a tracing function name from trace object code using a tracing framework, wherein the tracing function name comprises a probe handler;

determine an address location of the trace point in an instrumented program by accessing the probe handler in a trace point table; and

replace a placeholder function located at the address location of the trace point with a function call into the tracing framework.

(See rationale discussed in the rejection of claim 9 above)

As per Claims 29-38: Tamches discloses claims 29-38. See the rationale discussed in the rejection of Claims 10-19 above, respectively.

Conclusion

5. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Ted T. Vo whose telephone number is (571) 272-3706. The examiner can normally be reached on 8:00AM to 4:30PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Wei Y. Zhen can be reached on (571) 272-3708.

The facsimile number for the organization where this application or proceeding is assigned is the Central Facsimile number **571-273-8300**.

Any inquiry of a general nature or relating to the status of this application should be directed to the TC 2100 Group receptionist: 571-272-2100. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR

system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

TTV
July 20, 2007

TED VO
PRIMARY EXAMINER